

### **Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) A method for managing direct memory access resources, the method comprising:  
responsive to a change in context for a direct memory access resource, storing data relating to [[the]] a context switch in a context switch history containing a number of prior context switches occurring prior to a current context, wherein the context switch history is a circular context switch history table used for freeing portions of memory used for direct memory access chain of requests; and  
freeing portions of the direct memory access chain of requests using the context switch history to form freed portions, wherein the freed portions are reused for requests.
2. (Original) The method of claim 1, wherein the portions are direct memory access queues.
3. (Original) The method of claim 2, wherein the freeing step comprises:  
responsive to processing of requests, identifying a direct memory access queue for contexts prior to the current context to form an identified direct memory access queue; and  
freeing the identified direct memory access queue.
4. (Original) The method of claim 1, wherein the context switch history includes a pointer to a private structure for rendering the context, a pointer to a registration structure for pools of queues linked to the context, and an identifier for the thread.
5. (Canceled)
6. (Original) The method of claim 1, wherein the change in context occurs when the direct access memory resource is available by a first thread and access to the direct access memory resource is granted to a second thread and wherein the direct access memory resource is made available to the second thread by adding a buffer of the second thread to an end of a direct memory access chain of requests for the first thread to generate a direct memory access request for the second thread.

7. (Original) The method of claim 6, wherein the direct memory access request is a zero length direct memory access request.

8. (Original) The method of claim 2 further comprising:

responsive to encountering an error from a parameter in a request in the direct memory access chain of requests, identifying a queue originating the parameter in the direct memory access chain of requests using the context switch history, wherein a bad hardware address is identified using the queue.

9. (Currently Amended) A data processing system for managing direct memory access resources, the data processing system comprising:

storing means, responsive to a change in context for a direct memory access resource, for storing data relating to ~~[[the]]~~ a context switch in a context switch history containing a number of prior context switches occurring prior to a current context, wherein the context switch history is a circular context switch history table used for freeing portions of memory used for direct memory access chain of requests; and

freeing means for freeing portions of the direct memory access chain of requests using the context switch history to form freed portions, wherein the freed portions are reused for requests.

10. (Original) The data processing system of claim 9, wherein the portions are direct memory access queues.

11. (Original) The data processing system of claim 10, wherein the freeing means is a first freeing means and further comprising:

identifying means, responsive to processing of requests, for identifying a direct memory access queue for contexts prior to the current context to form an identified direct memory access queue; and  
second freeing means for freeing the identified direct memory access queue.

12. (Original) The data processing system of claim 10 further comprising:

identifying means, responsive to encountering an error from a parameter in a request in the direct memory access chain of requests, for identifying a queue originating the parameter in the direct memory access chain of requests using the context switch history, wherein a bad hardware address is identified using the queue.

13. (Currently Amended) A computer program product in a computer readable medium for managing direct memory access resources, the computer program product comprising:

first instructions, responsive to a change in context for a direct memory access resource, for storing data relating to ~~[[the]]~~ a context switch in a context switch history containing a number of prior context switches occurring prior to a current context, wherein the context switch history is a circular context switch history table used for freeing portions of memory used for direct memory access chain of requests; and

second instructions for freeing portions of the direct memory access chain of requests using the context switch history to form freed portions, wherein the freed portions are reused for requests.

14. (Original) The computer program product of claim 13, wherein the portions are direct memory access queues.

15. (Original) The computer program product of claim 14, wherein the second instructions comprises:

first sub-instructions, responsive to processing of requests, for identifying a direct memory access queue for contexts prior to the current context to form an identified direct memory access queue; and  
second sub-instructions for freeing the identified direct memory access queue.

16. (Original) The computer program product of claim 13, wherein the context switch history includes a pointer to a private structure for rendering the context, a pointer to a registration structure for pools of queues linked to the context, and an identifier for the thread.

17. (Canceled)

18. (Original) The computer program product of claim 13, wherein the change in context occurs when the direct access memory resource is available by a first thread and access to the direct access memory resource is granted to a second thread and wherein the direct access memory resource is made available to the second thread by adding a buffer of the second thread to an end of a direct memory access chain of requests for the first thread to generate a direct memory access request for the second thread.

19. (Original) The computer program product of claim 18, wherein the direct memory access request is a zero length direct memory access request.

20. (Original) The computer program product of claim 14 further comprising:  
third instructions, responsive to encountering an error from a parameter in a request in the direct memory access chain of requests, for identifying a queue originating the parameter in the direct memory access chain of requests using the context switch history, wherein a bad hardware address is identified using the queue.
21. (New) The method of claim 1, wherein each of the number of prior context switches in the context switch history is an entry in the context switch history, the entry comprising:  
a private value;  
a thread identifier;  
a process identifier;  
a save device buffer value; and  
a load device buffer value.
22. (New) The computer program product of claim 13, wherein each of the number of prior context switches in the context switch history is an entry in the context switch history, the entry comprising:  
a private value;  
a thread identifier;  
a process identifier;  
a save device buffer value; and  
a load device buffer value.